BS-PCC-3428

PC/104 工业计算机主板

硬件安装手册

手册版本：V1.0
适用板卡：VD1

地址：深圳市福田区车公庙天安数码城创新科技广场 B 座 1510
邮编：518040
电话：（0755） 83439980 83439280
传真：（0755） 83439680
网址：http://www.bsky.com.cn
邮箱：E-mail: market@bsky.com.cn

目　　录

1. 产品概述 ······ **1**
1.1 简介 ······ 1
1.2 特性 ······ 1
1.3 规格 ······ 1
1.4 附件清单 ······ 2

2. 接口介绍 ······ **3**
2.1 跳帽及连接器位置 ······ 3
2.2 跳帽及连接器简介 ······ 3
2.3 跳帽设置 ······ 4
2.4 接口引脚定义 ······ 4

3. 硬件安装 ······ **10**
3.1 注意事项 ······ 10
3.2 CRT/LCD 的连接 ······ 10
3.3 键盘/鼠标的连接 ······ 10
3.4 DOC/DOM/HDD 存储设备的连接 ······ 10
3.5 串口的设置和使用 ······ 10
3.6 其他端口的连接 ······ 11
3.7 电源输入 ······ 11
3.8 常见故障排除 ······ 11

4. BIOS 设置 ······ **12**
4.1 BIOS 设置 ······ 12
4.2 驱动程序安装 ······ 12

附录 ······ **13**
附录 1：系统资源（I/O，中断资源占用情况） ······ 13
附录 2：配线方法(sharp LQ104V1DJ11) ······ 14
附录 3：GPIO 编程实例 ······ 15

1. 产品概述

1.1 简介

PCC-3428 是一款在 90mm×96mm 尺寸上开发出来的全功能嵌入式工业计算机主板。该款主板采用 ST 公司的嵌入式 CPU STPC Atlas。Atlas CPU 是一款增强型 486 DX/DX2 CPU。它工作在 DX2 模式下的工作频率可达 133MHz。Atlas 内部集成了一个 2D 图形控制器和一个内存控制器，可以直接扩展 LCD/CRT 接口及 100MHz 的 SDRAM，最大容量可以支持到 64MBytes。PCC-3428 集成 10/100Mbps 网络接口、DOC 接口、EIDE、二串一并、二个 USB，并提供扩充用的 PC/104 接口。

主板采用 ST 公司高集成度的 STPC Atlas 芯片。 STPC Atlas 芯片除了集成一个 486 DX/DX2 的 CPU 内核以外，还集成了传统的南北桥和 SUPER I/O 芯片的功能，集成度高、工作稳定、低功耗 CPU，无需风扇，板上的 LCD 接口可以支持 TFT LCD 屏，分辨率最高达 1024×768。CRT 分辨率可达 1280×1024×75bpp。

PCC-3428 以其超全的功能、超小的体积，可广泛应用于各种嵌入式应用领域。如医疗器械、仪器仪表、网络终端、信息家电等各种领域。

1.2 特性

- ◆ 在板 STPC Atlas CPU，外频 66/100MHz，内频达 66/100/133MHz。
- ◆ 一个 10/100M 网络接口。
- ◆ 支持 CRT/LCD 显示接口。
- ◆ 支持 DOC 8MBytes～1GBytes。
- ◆ 在板 SDRAM，最大支持 64MBytes。
- ◆ 片上系统级芯片（SOC），性能稳定可靠。
- ◆ 低功耗 CPU，无需风扇。

1.3 规格

结构: 嵌入式 PC/104 工业计算机主板。
处理器: 在板STPC Atlas CPU，主频66MHz，可设为90MHz/100MHz/133MHz。
芯片组: 集成在Atlas芯片内。
系统内存: 在板32M SDRAM（64M可选）。
BIOS: Phoenix BIOS,即插即用。
显示接口： 支持CRT、18bit TFT LCD。
CRT分辨率最大支持1280×1024×75bpp；
LCD TFT最大支持1024×768×18bpp,支持VGA、SVGA、XGA、SXGA。
在板LAN： RealTek RTL8139C 10/100Mbps 以太网控制器。
EIDE接口： 支持标准硬盘和CD-ROM。
USB接口： 两个USB1.1接口。
Super I/O: 集成在Atlas芯片内。两个串口、一个并口、一个键盘/鼠标接口。
DOC接口： M-Systems flash盘，支持系统启动，容量8MBytes～1GBytes的存储能力。
总线支持： PC/104接口。
电源: 133MHz工作频率下，+5V@0.91A(最大)可单一电源+5V供电。
尺寸: 符合PC/104标准尺寸，90mm×96mm。
PCB板层数： 8层，抗电磁干扰能力强。
工作温度： 工业级-20℃～+70℃。
相对湿度： 相对湿度5%～95%，非凝结。
防EMI设计： 串口、并口、CRT接口、USB接口、键盘/鼠标接口防EMI设计。
串口特点: COM1支持外接RS485模块+5V供电，COM2可配置RS232/485输出。
看门狗定时器:16位可编程的主频分时，32位可编程的减数记数值。
GPIO接口： 16路GPIO口可按位设置为输入或输出。（该GPIO的描述要和3429统一起来）
特色推荐： 带网卡，LCD接口，2路USB，串口RS485/RS232模式切换免跳线。

1.4 附件清单

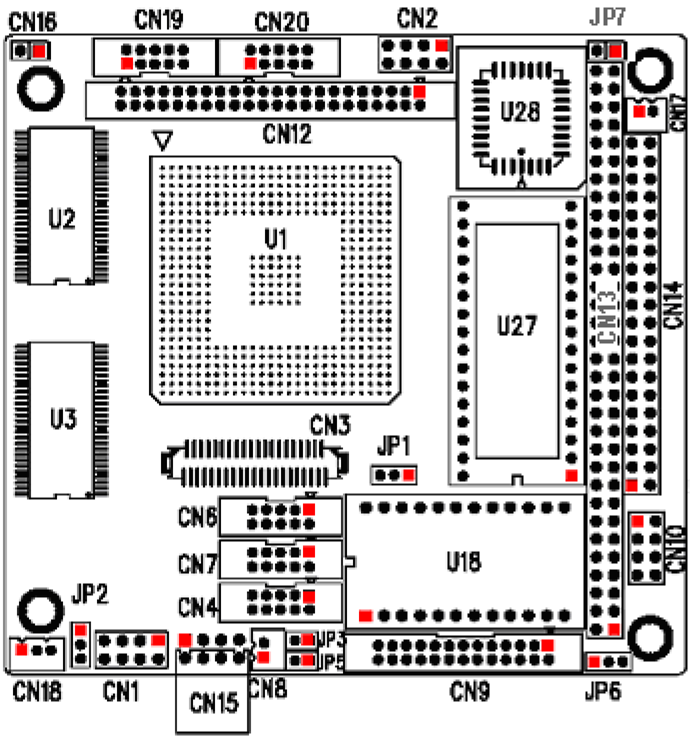
在安装您的CPU卡之前，请确认您是否收到了以下附件：

1. 1块PCC-3428 PC/104工业计算机主板。
2. 1块CD驱动程序碟片。(含本说明书)
3. 1根44pin IDE小硬盘线。
4. 2根DB9串口线。
5. 1根并口线。
6. 1根PS/2键盘鼠标二合一转接线。
7. 1根CRT转接线。
8. 1根USBx2转接线。

如附件不全或部分受损，请尽快与我们联系。以上附件仅为标准配置，可根据客户需要选配。

2. 接口介绍

2.1 跳帽及连接器位置



注意：该图用方块标示了所介绍的零件的第一引脚。

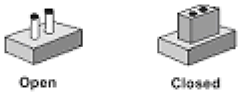
2.2 跳帽及连接器简介

跳帽位置	功　　　能
JP1	LCD 面板电压选择
JP2	LCD 背光电压选择
JP3	清除 CMOS
JP5	COM2 485 多机模式下挂接终端匹配电阻（150Ω）
JP6	CN6（COM1）引脚 4 输出+5V（外接 485 模块时使用）
JP7	BIOS 写保护跳线

接口位置	功　　　能
CN1	LAN 接口
CN2	USB 接口
CN3	LCD 数据线接口
CN4	CRT 接口
CN6	COM1 接口（可外接 485 模块）
CN7	COM2 接口
CN8	COM2 RS485 输出接口
CN9	并口/打印口
CN10	PS/2 键盘、鼠标接口
CN12	44PIN IDE 接口
CN13,CN14	PC/104 接口
CN15	外接电源接口
CN16	复位键接口
CN17	风扇接口（可提供系统风扇使用）
CN18	LCD 背光电源接口
CN19,CN20	GPIO 接口低 8 位/高 8 位

2.3 跳帽设置

跳帽设置示意图：



JP1：LCD 面板电压选择（参考后面 CRT/LCD 接线方法的介绍）

设　置	功　　能
Closed 1-2	面板电压为 5V
Closed 2-3	**面板电压为 3.3V（默认）**

JP2：LCD 背光电压选择（参考后面 CRT/LCD 接线方法的介绍）

设　置	功　　能
Closed 1-2	**背光电压为 5V（默认）**
Closed 2-3	背光电压为 12V

JP3:清除 CMOS

设　置	功　　能
Open	**正常工作时（默认）**
Close	清除 CMOS

JP5：COM2 485 模式终端电阻设定（参考后面串口使用说明）

设　置	功　　能
Open	**不加终端电阻（默认）**
Close	加终端电阻

JP6：设定 CN6（COM1）引脚 4 输出（参考后面串口使用说明）

设　置	功　　能
Closed 1-2	由 CN6 的 4 脚提供+5V 供外接 RS485 模块用
Closed 2-3	**由 CN6 的 4 脚提供 RS232 电平的 DTR 信号(默认)**

JP7：BIOS 写保护跳线

设　置	功　　能
Open	**写保护（默认）**
Close	写允许，可在板烧录 BIOS

2.4 接口引脚定义

CN1：LAN 接口（下表括号内为对应的网线颜色）

信 号 名	引 脚	引 脚	信 号 名
TX+（白绿）	1	2	TX-（绿）
RX+（白蓝）	3	4	NC （蓝）
NC （白红）	5	6	RX-（红）
NC （白棕）	7	8	NC （棕）

CN2：USB 接口（接 USBx2 转接线）

USB1		USB2	
信 号 名	引 脚	引 脚	信 号 名
VCC	1	2	VCC
USBD1-	3	4	USBD2-
USBD1+	5	6	USBD2+
GND	7	8	GND

CN3：LCD 数据线接口

信 号 名	引 脚	引 脚	信 号 名
Clock	1	2	GND
GND	3	4	D5（B5）
D17（R5）	5	6	D4（B4）
D16（R4）	7	8	D3（B3）
D15（R3）	9	10	backlight enable
D14（R2）	11	12	NC
GND	13	14	D2（B2）
D13（R1）	15	16	D1（B1）
D12（R0）	17	18	D0（B0）
D11（G5）	19	20	GND
D10（G4）	21	22	ENAB
GND	23	24	VCC
D9（G3）	25	26	VCC
D8（G2）	27	28	VSYNC
D7（G1）	29	30	GND
D6（G0）	31	32	HSYNC
GND	33	34	VCC
GND	35	36	VCC
GND	37	38	GND
GND	39	40	GND
NC	41		

CN4：CRT 接口（CRT 转接线）

信 号 名	引 脚	引 脚	信 号 名
RED	1	2	AGND
GREEN	3	4	GND
BLUE	5	6	AGND
VSYNC	7	8	DDCD
HSYNC	9	10	DDCK

CRT 接口接 CRT 转接线，CRT 非一对一转接。CRT 转接线的另一头（接显示器的一头）的引脚定义如下：

信 号 名	引 脚	引 脚	信 号 名
Red	1	2	GND
NC	3	4	Green
GND	5	6	DDCSDA
Blue	7	8	GND
HSYNC	9	10	NC
VCC	11	12	VSYNC
GND	13	14	GND
DDCSCL	15		

CN6：COM1 接口（接 DB9 串口线）

信 号 名	引 脚	引 脚	信 号 名
DCD 数据载波检测	1	2	RXD 接收数据
TXD 发送数据	3	4	DTR 或+5V （参见 JP6）
GND 信号地	5	6	DSR 数据设备准备好
RTS 请求发送	7	8	CTS 清除发送
RI 振铃指示	9	10	NC 未接

CN7：COM2 接口（接 DB9 串口线）

信 号 名	引 脚	引 脚	信 号 名
DCD 数据载波检测	1	2	RXD 接收数据
TXD 发送数据	3	4	DTR 数据终端准备好
GND 信号地	5	6	DSR 数据设备准备好
RTS 请求发送	7	8	CTS 清除发送
RI 振铃指示	9	10	NC 未接

CN8：COM2 RS485 输出接口（参考后面串口使用说明）

引 脚	信 号 名
1	485+
2	485-

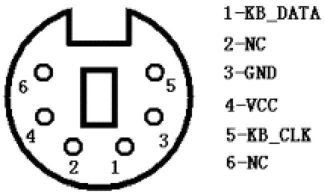
CN9：并口/打印口（接打印机并口线）

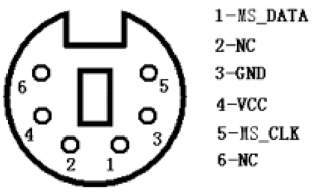
信 号 名	引 脚	引 脚	信 号 名
STB	1	2	Auto Feed
PD0	3	4	Error
PD1	5	6	Initialize
PD2	7	8	Select IN
PD3	9	10	GND
PD4	11	12	GND
PD5	13	14	GND
PD6	15	16	GND
PD7	17	18	GND
ACK	19	20	GND
Busy	21	22	GND
Paper Empty	23	24	GND
Select	25	26	GND

CN10：PS/2 键盘/鼠标接口（接 PS/2 键盘鼠标二合一转接线）

信 号 名	引 脚	引 脚	信 号 名
KBDATA	1	2	MSDATA
KBCLK	3	4	MSCLK
GND	5	6	GND
+5V	7	8	+5V

PS/2 键盘鼠标二合一转接线的另外一端的信号定义为：

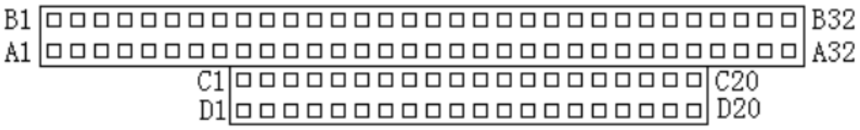




CN12: 44Pin IDE 接口（接 44pin IDE 小硬盘线）

信 号 名	引 脚	引 脚	信 号 名
Reset	1	2	GND
Data 7	3	4	Data 8
Data 6	5	6	Data 9
Data 5	7	8	Data 10
Data 4	9	10	Data 11
Data 3	11	12	Data 12
Data 2	13	14	Data 13
Data 1	15	16	Data 14
Data 0	17	18	Data 15
GND	19	20	NC
DRQ0	21	22	GND
HD IOW	23	24	GND
HD IOR	25	26	GND
IOCHRDY	27	28	GND
DACK0	29	30	GND
IRQ14	31	32	/IOCS16
Address 1	33	34	NC
Address 0	35	36	Address 2
CS1	37	38	CS3
Active LED	39	40	GND
VCC	41	42	VCC
GND	43	44	NC

CN13、CN14:PC/104 长/短接口



CN13				CN14			
引脚	信号名	引脚	信号名	引脚	信号名	引脚	信号名
A1	IOCHK	B1	GND				
A2	D7	B2	RESET				
A3	D6	B3	VCC				
A4	D5	B4	IRQ9				
A5	D4	B5	-5V				
A6	D3	B6	DRQ2				
A7	D2	B7	-12V				
A8	D1	B8	ZWS				
A9	D0	B9	+12V	C1	GND	D1	GND
A10	IOCHRDY	B10	GND	C2	SBHE	D2	MEMCS16
A11	AEN	B11	SMEMW	C3	LA23	D3	IOCS16
A12	A19	B12	SMEMR	C4	LA22	D4	IRQ10
A13	A18	B13	IOW	C5	LA21	D5	IRQ11
A14	A17	B14	IOR	C6	LA20	D6	IRQ12
A15	A16	B15	DACK3	C7	LA19	D7	IRQ15
A16	A15	B16	DRQ3	C8	LA18	D8	IRQ14
A17	A14	B17	DACK1	C9	LA17	D9	DACK0
A18	A13	B18	DRQ1	C10	MEMR	D10	DRQ0
A19	A12	B19	REFRESH	C11	MEMW	D11	DACK5
A20	A11	B20	CLK	C12	D8	D12	DRQ5
A21	A10	B21	IRQ7	C13	D9	D13	DACK6
A22	A9	B22	IRQ6	C14	D10	D14	DRQ6

A23	A8	B23	IRQ5	C15	D11	D15	DACK7
A24	A7	B24	IRQ4	C16	D12	D16	DRQ7
A25	A6	B25	IRQ3	C17	D13	D17	VCC
A26	A5	B26	DACK2	C18	D14	D18	MASTER
A27	A4	B27	TC	C19	D15	D19	GND
A28	A3	B28	BALE	C20	KEY PIN	D20	GND
A29	A2	B29	VCC				
A30	A1	B30	OSC				
A31	A0	B31	GND				
A32	GND	B32	GND				

CN15: 电源接口（电源输入接口，参考后面详细介绍）

引 脚	信 号 名
1	+12V
2	GND
3	GND
4	+5V

CN16:复位键接口（接复位开关）

设 置	功　　能
1	RESET
2	GND

CN17:风扇接口（接系统风扇）

引 脚	信 号 名
1	+12V
2	GND

CN18: LCD 背光电压接口（参考 JP2 的设置）

引 脚	信 号 名
1	经 PWM 调节的+12V（或+5V）输出
2	保留
3	GND

CN19: 8 位 GPIO 接口（与 CN20 一起构成 16 位 GPIO 的低 8 位）

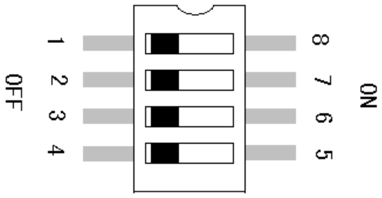
引 脚	信 号 名
1	D0（16 位 GPIO 的 D0）
2	D1（16 位 GPIO 的 D1）
3	D2（16 位 GPIO 的 D2）
4	D3（16 位 GPIO 的 D3）
5	D4（16 位 GPIO 的 D4）
6	D5（16 位 GPIO 的 D5）
7	D6（16 位 GPIO 的 D6）
8	D7（16 位 GPIO 的 D7）
9	GND
10	GND

CN20: 8 位 GPIO 接口（与 CN19 一起构成 16 位 GPIO 的高 8 位）

引 脚	信　号　名
1	D0（16 位 GPIO 的 D8）
2	D1（16 位 GPIO 的 D9）
3	D2（16 位 GPIO 的 D10）
4	D3（16 位 GPIO 的 D11）
5	D4（16 位 GPIO 的 D12）

6	D5（16 位 GPIO 的 D13）
7	D6（16 位 GPIO 的 D14）
8	D7（16 位 GPIO 的 D15）
9	GND
10	GND

SW1：设置 DOC 的内存占用空间。



1-8	2-7	3-6	4-5	DOC 占用内存空间
ON	ON	ON	ON	(不可用保留)
ON	ON	ON	OFF	(不可用保留)
ON	ON	OFF	ON	C000:8000 - C000:9FFF
ON	ON	OFF	OFF	C000:C000 - C000:DFFF
ON	OFF	ON	ON	D000:0000 - D000:1FFF
ON	OFF	ON	OFF	D000:4000 - D000:5FFF
ON	OFF	OFF	ON	D000:8000 - D000:9FFF
ON	**OFF**	**OFF**	**OFF**	**D000:C000 - D000:DFFF(默认)**
OFF	ON	ON	ON	(不可用保留)
OFF	ON	ON	OFF	(不可用保留)
OFF	ON	OFF	ON	(不可用保留)
OFF	ON	OFF	OFF	(不可用保留)
OFF	OFF	ON	ON	(不可用保留)
OFF	OFF	ON	OFF	(不可用保留)
OFF	OFF	OFF	ON	(不可用保留)
OFF	OFF	OFF	OFF	(不可用保留)

3. 硬件安装

3.1 注意事项

注意：请在断电条件下插拔部件；在连接电源接头到主板前请先确认电源处于关闭状态，以避免瞬间的电源冲击造成敏感元件的损坏。

从包装盒中取出主板后，请确认没有由于运输过程造成的针脚或其它短路！

小心：现代电子产品对静电非常敏感，拿主板前，请戴上静电手环或静电手套以将您身上的静电导走。请将主板放置在静电桌垫或静电袋内。

使用前请详细阅读本安装手册，确认主板的跳线配置正确，若因您的误操作造成产品损坏，请恕不负赔偿责任。

3.2 CRT/LCD的连接

PCC-3428 主板可同时支持 CRT 和 18bit 彩色 TFT LCD 显示屏输出而不相互影响，使用时可以同时接上使用。附件中有 CRT 转接线，将转接线的一头接到主板的 CN4，另一头接 CRT 显示器，CRT 上电即可正确显示。

在使用 LCD 之前，请先确认您的 LCD 屏是 3.3V 还是 5V 的屏（这指的是您使用的 LCD 屏里面的 IC 的工作电压为 3.3V 还是 5V。请咨询您的 LCD 屏供应商，市面上大多数屏为 3.3V）。您可以通过设置 JP1 来改变 CN3 的输出电压，当 JP1 设定为 3.3V 或 5V 时，对应的 CN3 的 VCC 脚将会输出 3.3V 或 5V 电压，所以在使用前务必先设定正确的 LCD 屏工作电压。

LCD 屏的数据线一头接 CN3，另一头接 LCD 显示屏。LCD 一般还需要提供背光电源，使用前请确认您的 LCD 屏的背光电源为+5V 还是+12V，您可以从外部供电或从主板的 CN18 取电，从主板取电时，请设定正确的电压输出（JP2），并确保 CN18 的引脚定义和您的 LCD 屏的定义一致（参考 CN18 的引脚定义）。

不同的 LCD 屏有不一样的接口定义，我们在附件中没有数据线，您需要找您的 LCD 供应商根据我们主板上 LCD 接口定义来为您配您所使用的 LCD 屏的数据线。

3.3 键盘/鼠标的连接

附件中有一根 PS/2 键盘鼠标二合一转接线，将它接到 CN10，这样就可以转接出标准的 PS/2 键盘鼠标接口。插 CN10 时，需要注意方向。键盘鼠标二合一转接线用白色线标示第一脚，连接时，白色线朝上对准 CN10 的第一脚接上即可。

3.4 DOC/DOM/HDD存储设备的连接

该主板支持 DOC/DOM/HDD 设备，DOC 插入 DOC 座时请务必注意缺口对缺口，否则将烧毁您的 DOC 盘。DOM/HDD 接到 CN12。DOM 请选用 44PIN 卧式的，HDD 请选用小硬盘，否则您可能需要额外准备转接线。您也可以选择我们公司设计的 44PIN IDE 接口转 CF 卡接口，采用 CampactFlash 卡来取代传统的硬盘，将您的操作系统及应用程序装到 CampactFlash 卡中。

3.5 串口的设置和使用

该款主板含两个标准的串口 COM1，COM2。它们被设计成一定的灵活性，以满足您的实际需求。

COM1：标准的 RS232 信号从 CN6 输出，其中 CN6 的信号可以通过设置 JP6 重新配置。当 JP6 设置为 1-2 时，可以从 CN6.4 输出+5V 的电源，这在需要通过串口来扩展外部模块（如 RS232/RS485/RS422 转换器，或这通过该口来与单片机通讯时，可以不要给外部模块额外供电），这个时候，您可能无法使用 DTR 信号。如果您要使用 DTR，这时您要将 JP6 设置 2-3，这样您的外部模块要另外供电。

COM2：支持标准的 RS232 信号或 RS485 信号输出。选择 RS232 或 RS485，您无需作任何设置。使用 RS232 信号接到 CN7，使用 RS485 信号接到 CN8（注意，使用时只能选择其一，不能同时接）。

RS485 为半双工模式，所以选择 RS485 模式时，需要软件上切换 RS485 线的收或发的模式。这可以通过控

制相应的寄存器的值来使 DTR 信号为高或为低。当 DTR 信号为高电平时，RS485 为接收模式，当 DTR 信号为低电平时，RS485 为发送模式。

RS485 常用在总线型多机通讯网络中，当多机通讯时，且传输距离较远时，位于两端的两台 PC 必须设置终端电阻以实现正确传输。需要设置终端电阻时，您只要将 JP5 设置为 Close 状态。

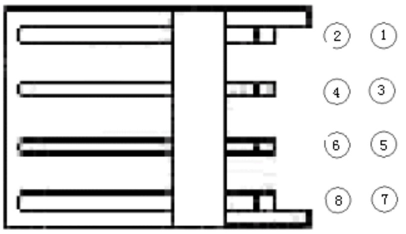
3.6 其他端口的连接

复位开关：　接复位开关线到 CN16，无方向。
系统风扇：　CN17，有缺口方向。
并　　口：　接打印机并口线到 CN9，有缺口方向。
USB　 口：　接 USBx2 转接线到 CN2，白线朝下。
GPIO：　　　TTL 电平的数字输入/出，接到 CN19，CN20，有缺口方向。
PC/104 接口：需要时可以采用该接口扩展 PC/104 模块。

3.7 电源输入

该款主板的电源接口定义默认为与标准的软驱接口一致。为保证该板的兼容性，PCC-3428 的电源输入部分可以提供+12V、+5V、-12V、-5。默认只提供+12V、+5V，当您需要使用-12V、-5V 时，取下 CN15，自己重新配线。

取下 CN15 后，您会发现板子上有 8 个接口，其定义如下：



-5V	2	1	+12V
GND	4	3	GND
GND	6	5	GND
-12V	8	7	+5V

PCC-3428 支持单+5V 供电。选择单+5V 供电时，首先确认您的系统没有扩展 PC/104 模块，或扩展了 PC/104 模块，但模块没有使用+12V 电源。

单+5V 供电时，PC/104 接口，CN17 风扇接口，CN18 LCD 背光电源接口将无法提供+12V 电源，这意味着，不支持使用了+12V 的 PC/104 模块，不支持需要+12V 背光的 LCD 屏，不能接系统风扇。如果您没有使用上述功能，请大胆使用单一+5V 供电。

我们提供的是标准的软驱接口一致的电源引脚定义，这方便您在做样机时使用电脑的电源，当您需要实际生产时，建议您根据实际情况，选择方便的连接头或直接焊线。配线时，请注意线的粗细，应该保证能通过 1.6A 的电流而不烧毁。

3.8 常见故障排除

正确接线情况下，系统 BIOS 会执行自检，并会显示相关内容。出现故障时请确认：

1. 主板是否设置正确。
2. 电源是否正确接入。
3. 是否有显示。
4. BIOS 是否正确设置。
5. 操作系统是否工作正常。
6. 是无法启动系统还是局部功能工作有问题。
7. 确认您的应用软件有无问题（使用第三方软件来测试硬件）。

您可以采用替代法，如更换主板、更换电源等方法来推断问题出在哪里；您也可将上述问题及时反映给我们，我们一定能给您满意的答复。

4. BIOS 设置

4.1 BIOS设置

BIOS 是电脑软件启动的开始点，BIOS 可以将您的硬件配置成您所需要的设定。正确设置 BIOS 是每个电脑工程师的基本技能，本说明书对 BIOS 设置不做太深的探讨，请参考相关资料。

PCC-3428 支持 DOS、Win3.1、Win95、Win98、Linux、WinCE 等支持 i386 的操作系统。部分操作系统下的使用，需要参考相关的操作系统类的资料。

PCC-3428 常采用嵌入式系统来满足不同的应用领域。较常用的嵌入式系统如：Linux、WinCE，PCC-3428 都能提供较好的支持。嵌入式系统可以存储在 DOC（U27）或者 DOM（CN12）中。

4.2 驱动程序安装

随PCC-3428主板附的光碟含有您使用它可能用到的所有驱动程序和应用软件。

在不同的操作系统下的驱动安装会有所不同。

我们建议您在使用主板时，正确安装驱动程序。

如果您使用Windows NT系列的操作系统，我们强烈建议您到微软的网站下载最新的Service Pack。

Windows 9x/ME/NT 4.0的驱动安装

1. 插入PCC-3428 光碟到CD-ROM 驱动器。
2. 点击开始/设置/控制面板。
3. 双击添加硬件，会弹出添加硬件向导，按下一步。
4. 选择搜索比当前设备使用的驱动程序更好的驱动程序，按下一步。
5. 选择指定位置，点击浏览。找到光碟对应操作系统下的驱动程序目录，按确定，按下一步。
6. 向导提示您找到了所要的驱动程序，按下一步。
7. 系统提示您插入Windows 98光碟，插入光碟，按确认键。
8. 按完成，重新启动系统。

附录

附录1：系统资源（I/O，中断资源占用情况）

标准设备的I/O地址表，部分未用或可以通过BIOS禁用，开发扩展卡请尽量避开常用的I/O地址:

地 址	设 备
000 - 01F	DMA控制器#1
020 - 03F	中断控制器#1
040 - 05F	定时器
060 - 06F	键盘控制器
070 - 07F	实时时钟NMI
080 - 09F	DMA 页面寄存器
0A0 - 0BF	中断控制器#2
0C0 - 0DF	DMA 控制器#2
0F0 - 0F1	清/复位数学协处理器
1F0 - 1F7	硬盘控制器
200 - 210	游戏端口
278 - 27F	并口#2（本板未占用）
2E8 - 2EF	串口#4(COM4，本板未占用)
2F8 - 2FF	串口#2(COM2)
300 - 31F	原型卡/流线型磁带适配器
360 - 36F	PC网络
378 - 3FF	并口#1
380 - 38F	SDLC #2
3A0 - 3AF	SDLC #1
3B0 - 3BF	MDA视频卡(包含LPT0)
3C0 - 3CF	EGA卡
3D0 - 3DF	CGA卡
3E8 - 3EF	串口#3(COM3，本板未占用)
3F0 - 3F7	软磁盘控制器（CMOS中可关闭）
3F8 - 3FF	串口#1(COM1)

中断资源

软中断号	硬中断线	描 述	中断向量地址
0x08	IRQ0	定时器输出	0x00000：0x00020
0x09	IRQ1	键盘	0x00000：0x00024
NOT ACTIVE	IRQ2	中断级联	
0x0B	IRQ3	串口#2	0x00000：0x0002C
0x0C	IRQ4	串口#1	0x00000：0x00030
0x0D	IRQ5	未使用	0x00000：0x00034
0x0E	IRQ6	未使用	0x00000：0x00038
0x0F	IRQ7	并口#1	0x00000：0x0003C
0x70	IRQ8	实时时钟	0x00000：0x001C0
0x71	IRQ9	未使用	0x00000：0x001C4
0x72	IRQ10	未使用	0x00000：0x001C8
0x73	IRQ11	未使用	0x00000：0x001CC
0x74	IRQ12	未使用	0x00000：0x001D0
0x75	IRQ13	数学协处理器	0x00000：0x001D4
0x76	IRQ14	主硬盘	0x00000：0x001D8
0x77	IRQ15	未使用	0x00000：0x001DC

附录2：配线方法(sharp LQ104V1DJ11)

屏引脚	信号名称	信号定义	主板引脚
1	GND	GND	2
2	CK	Clock signal for sampling each data signal	1
3	Hsync	Horizontal synchronous signal	32
4	Vsync	Vertical synchronous signal	28
5	GND	GND	3
6	R0	R E D data signal (LSB)	17
7	R1	R E D data signal	15
8	R2	R E D data signal	11
9	R3	R E D data signal	9
10	R4	R E D data signal	7
11	R5	R E D data signal (MSB)	5
12	GND	GND	13
13	G0	G R E E N data signal(LSB)	31
14	G1	G R E E N data signal	29
15	G2	G R E E N data signal	27
16	G3	G R E E N data signal	25
17	G4	G R E E N data signal	21
18	G5	G R E E N data signal(MSB)	19
19	GND	GND	23
20	B0	B L U E data signal(LSB)	18
21	B1	B L U E data signal	16
22	B2	B L U E data signal	14
23	B3	B L U E data signal	8
24	B4	B L U E data signal	6
25	B5	B L U E data signal(MSB)	4
26	GND	GND	20
27	ENAB	Signal to settle the horizontal display position	22
28	Vcc	+5.0V power supply	24
29	Vcc	+5.0V power supply	26
30	R/L	Horizontal display mode select signal	
31	U/D	Vertical display mode select signal	

附录3：GPIO编程实例

本产品集成了 16 位的 GPIO，每 1 位可以定义成输入或输出。16 位的 GPIO 分为 2 组，每组 8 位。对 GPIO 的操作是按组进行的，每组 GPIO 对应 8 个控制或数据寄存器。第 1 组 GPIO 的 8 个寄存器对应的 I/O 地址是 0x320~0x327；第 2 组 GPIO 的 8 个寄存器对应的 I/O 地址是 0x328~0x32F。

对 GPIO 的操作除了其本身的控制或数据寄存器外，还必须设置南桥中有关寄存器。这些寄存器定义 GPIO 设备的开启和关闭、GPIO 的基地址及有关中断的设置。

下面给出一个示例：

```
/**********************************************************
*            蓝天 GPIO  演示程序
*
*   说明：
*   1）本程序适用于蓝天 PCC-3428  和  PCC-3429
*   2）用本程序进行测试时，请将第 1 组 GPIO
*       与第 2 组 GPIO 进行互连。
*   3）虽然可以对 GPIO 中的每 1 位进行操作，但为了方便说明，
*       本程序对组（8 位）进行操作。
*   4）函数 iowrite32（）和 ioread32()分别用于 32 位寄存器的
*       读和写操作。因为南桥中寄存器都是 32 位。
*
*   Version:   2.0
*   Author:     Xu Houqing
*   Date:       8/4/2005
*
**********************************************************/

#include <stdio.h>
#include <conio.h>

#define PCICMD 0xcf8      //PCI 地址索引寄存器
#define PCIDAT 0xcfc      // PCI 数据寄存器

extern void iowrite32( unsigned long address, unsigned long value );
extern unsigned long ioread32( unsigned long address );

//32 位寄存器的写操作函数
void iowrite32( unsigned long address, unsigned long value )
{
    asm{
        db     66h
        mov dx, word ptr address
        db     66h
        mov ax, word ptr value
        db     66h
        out    dx, ax
    }
```

```
}


//32 位寄存器的读操作函数
unsigned long ioread32( unsigned long address )
{
        unsigned long value;

        asm{
                db    66h
                push dx
                push ax
                mov dx, word ptr address
                db    66h
                in     ax, dx
                db    66h
                pop          ax
                pop  dx
        }
}

//设置南桥中 0x40 号寄存器，  开启 GPIO 设备（开机时处于关闭状态）
void enable_gpio()
{
        int value;

                // PCI 地址索引寄存器位定义：
                // bit 31        Enable
                // bit 30-24 Reserved
                // bit 23-16 Bus number
                // bit 15-11 Device number
                // bit 10-8    Function number
                // bit 7-2      Register number
                // bit 1-0      Reserved(always 0)

                iowrite32(PCICMD, 0X80006040L );     //索引到南桥中 0x40 号寄存器的

                iowrite32( PCIDAT, 0x00000002L );   //写入 0x00000002 表示开启 GPIO
}

int main()
{
        unsigned char i;
        int PORTA=0x320; //第 1 组 GPIO 寄存器基地址
        int PORTB=0x328; //第 2 组 GPIO 寄存器基地址

        unsigned char value1, value2;
```

```
enable_gpio(); //开启 GPIO 设备

printf( "\r\n" );
printf( "   Direct     Output      Input            Result \r\n" );
printf( " =======+==========+===========+======= " );


//先将第 1 组 GPIO 作为输出、第 2 组 GPIO 作为输入进行测试
outportb( PORTA, 0x00 );      //设置第 1 组 GPIO 作为输出
outportb( PORTB, 0xff );       //设置第 2 组 GPIO 作为输入

for ( i=0; i<8; i++ )
{
   value1 = 1<<i;

   outportb( PORTA+6, value1 ); //将数据输出到第 1 组 GPIO

   value2=inportb( PORTB+6);    //从第 2 组 GPIO 输入数据

   printf( "\r\n    A->B    |    PortA=%2x    PortB=%2x    | ", value1, value2 );

   //因为测试时两组 GPIO 互连，输入值应该等于输出值
   if ( value1==value2 )
      printf( " PASS " );
   else
      printf( " FAIL " );

}

printf( "\r\n    ----         --------    --------         ---- " );


//再将第 2 组 GPIO 作为输出、第 1 组 GPIO 作为输入进行测试
outportb( PORTA, 0xff );       //设置第 1 组 GPIO 作为输入
outportb( PORTB, 0x00 );      //设置第 2 组 GPIO 作为输出

for ( i=0; i<8; i++ )
{
   value1 = 1<<i;

   outportb( PORTB+6, value1 ); //将数据输出到第 2 组 GPIO

   value2=inportb( PORTA+6);    //从第 1 组 GPIO 输入数据

   printf( "\r\n    B->A    |    PortB=%2x    PortA=%2x    | ", value1, value2 );
```

```
        //因为测试时两组 GPIO 互连，输入值应该等于输出值
        if ( value1==value2 )
          printf( " PASS " );
        else
          printf( " FAIL " );

    }

    printf("\r\n\r\n");

}

/*  程序执行结果显示

  Direct    Output      Input          Result
 =======+==========+==========+=======
  A->B  |  PortA= 1   PortB= 1   |  PASS
  A->B  |  PortA= 2   PortB= 2   |  PASS
  A->B  |  PortA= 4   PortB= 4   |  PASS
  A->B  |  PortA= 8   PortB= 8   |  PASS
  A->B  |  PortA=10   PortB=10   |  PASS
  A->B  |  PortA=20   PortB=20   |  PASS
  A->B  |  PortA=40   PortB=40   |  PASS
  A->B  |  PortA=80   PortB=80   |  PASS
  ----      --------   --------      ----
  B->A  |  PortB= 1   PortA= 1   |  PASS
  B->A  |  PortB= 2   PortA= 2   |  PASS
  B->A  |  PortB= 4   PortA= 4   |  PASS
  B->A  |  PortB= 8   PortA= 8   |  PASS
  B->A  |  PortB=10   PortA=10   |  PASS
  B->A  |  PortB=20   PortA=20   |  PASS
  B->A  |  PortB=40   PortA=40   |  PASS
  B->A  |  PortB=80   PortA=80   |  PASS
*/
```